

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-70890

(43)公開日 平成7年(1995)3月14日

(51) Int.Cl.⁶

識別記号 庁内整理番号
7152-3B

F I

技術表示箇所

審査請求 有 請求項の数 4 FD (全 18 頁)

(21) 出願番号 特願平5-240714

(22)出願日 平成5年(1993)8月31日

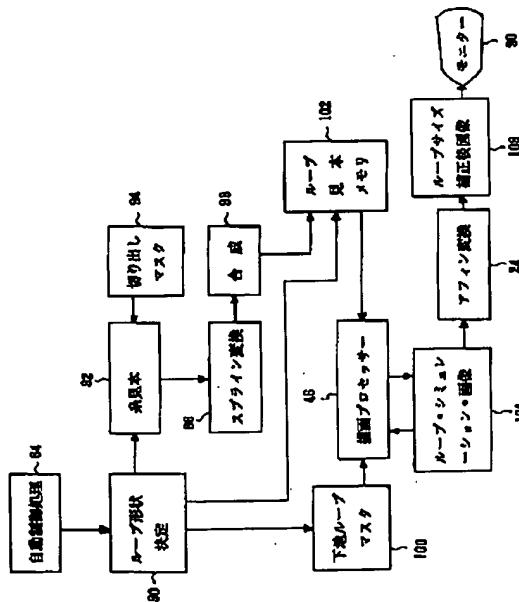
(71)出願人 000151221
株式会社島精機製作所
和歌山県和歌山市坂田85番地
(72)発明者 南 潔
和歌山県和歌山市有本555番19号
(74)代理人 弁理士 塩入 明 (外1名)

(54) 【発明の名称】 ニットデザインシステム

(57) 【要約】

【目的】 編物やループ見本を実際に編みます、編物をループを正確に表現するようにシミュレーションする。

【構成】 自動制御処理手段 6.4 で発生した編機用データをループ形状決定手段 9.0 で解析し、ループの形状、位置、糸の種類、ループ各部の明暗を分析し、糸見本記憶手段 9.2 からループ各部の明暗に応じたセグメントを切り出し、セグメントを合成して、ループ見本を作成する。次に下地ループとの重なり具合をマスクで表現し、フレームメモリ 10.4 に書き込み、アフィン変換手段 2.4 でループのアスペクト比を補正し、モニター 3.0 に表示する。



(2)

特開平7-70890

1

【特許請求の範囲】

【請求項1】 編機用データを基に、各ループの糸の種類とループ形状、ループ位置、ループ各部の明暗を決定し、かつ下地のループとの重なりを決定するためのループ形状決定手段と、各糸毎に少なくとも1つの糸見本を記憶するための糸見本記憶手段と、ループ形状決定手段のデータにより、各ループを複数のセグメントに分割し、糸見本記憶手段から糸見本を取り出してセグメントとし、各セグメントにループ各部の明暗に応じた明暗を与える、かつループ形状に応じてセグメントを屈曲させて、屈曲後のセグメントを合成してループ見本を形成するための、ループ見本作成手段と、編物をシミュレーションするように、ループ見本をループ形状決定手段のデータにより定めた位置に記憶して、複数のループを記憶するためのフレームメモリと、ループ形状決定手段のデータにより、下地のループとの重なりを表すマスクを作成するためのマスク作成手段と、前記マスクを用いて、前記フレームメモリに、下地ループの露出部を残すように、ループ見本を書き込むための描画手段と、前記フレームメモリの画像を表示するためのモニター、とを設けたことを特徴とする、ニットデザインシステム。

【請求項2】 前記フレームメモリの画像をループのアスペクト比に応じて補正する手段を設けて、アスペクト比の補正後の画像をモニターに表示させるようにしたことを特徴とする、請求項1のニットデザインシステム。

【請求項3】 ループ見本を記憶するための手段を設けて、出現頻度の高いループ見本を記憶するようにしたことを特徴とする、請求項1のニットデザインシステム。

【請求項4】 マネキンを表す画像を記憶するための第2のフレームメモリと、メッシュマッピング手段とを設けて、

第1のフレームメモリの編物のシミュレーション画像を、第2のフレームメモリの画像にメッシュマッピングするようにしたことを特徴とする、請求項2のニットデザインシステム。

【発明の詳細な説明】

【0001】

【発明の利用分野】 この発明は、編機用データを基に編物をシミュレーションするためのニットデザインシステムに関し、特にループ形状をシミュレーションするようにしたデザインシステムに関する。なおこの明細書において、編物とはセーター等の形態を伴った編物のみでなく、特定の形態の無い編地をも意味するものとする。

【0002】

【従来技術】 出願人は、モニター上で編物をデザインするシステムを提案した(特公平3-21661号)。こ

2

のシステムではデザイン過程の編物のデータをフレームメモリに記憶し、例えば1ループに1ピクセルを割り当てる、ピクセルの色で柄等の組織データを記憶する。例えば色彩の自由度を256とすれば、ループの編成方法について、256種のデータを表現できる。使用する色糸の種類はオプションで指定し、組織データの組織の種類毎に指定する。またこのシステムでは、デザイン終了後のフレームメモリのデータを、参照表等を用いて編機で編めるデータに変換する。

【0003】 次に考えられることは、編物を実際に編むことなく、モニター上やプリンタ上に編物をシミュレーションした画像を発生させることである。このことが実現されれば、編機を動かすことなしに、デザインした編物のサンプルを直ちに発生させ、短期間で多数のデザインをし、評価することが可能になる。またシミュレーション画像は、ループ形状だけでなく、各ループをその明暗や重なり具合と共に表現できる程度の精密さが必要である。

【0004】 これに対してまず考えられることは、ループの見本を実際に編んで用意し、スキャナー等で読み取り、フレームストアにスタンプすることである。しかしながらループ見本は多種多様であり、全てのループ見本を実際に用意することは困難である。例えば16種の糸を用いる場合、ループの重なりを考慮すると、ループ見本は少なくとも256種必要になる。そしてこれに組織形状を加えると、必要なループ見本の数は極端に増加する。そのためスキャナーでループ見本を読み込む手法では、代表的なループのみをシミュレーションできることになり、シミュレーションの精度が不足する。

【0005】

【発明の課題】 この発明の課題は、以下の点にある。

- 1) ループ見本を実際に編んで作成せずに、各ループを正確に表現するようにシミュレーションできるようにすること(請求項1~4)。
- 2) 各ループを、その形状、屈曲の具合い、ループ各部の明暗、他のループとの重なり具合いを含めてシミュレーションできるようにすること(請求項1~4)。
- 3) 編機用データをチェックし、実際に編めるデータかどうかを確認できるようにすること(請求項1~4)。
- 4) ループのアスペクト比を補正し、実際の編物に近い形状でシミュレーションできるようにすること(請求項2, 4)。
- 5) シミュレーション時間を短縮すること(請求項3)。
- 6) シミュレーションした画像をマネキンに試着させ、立体感のあるシミュレーションを行うこと(請求項4)。

【0006】

【発明の構成】 この発明は、編機用データを基に、各ループの糸の種類とループ形状、ループ位置、ループ各部

(3)

特開平7-70890

3

の明暗を決定し、かつ下地のループとの重なりを決定するためのループ形状決定手段と、各糸毎に少なくとも1つの糸見本を記憶するための糸見本記憶手段と、ループ形状決定手段のデータにより、各ループを複数のセグメントに分割し、糸見本記憶手段から糸見本を取り出してセグメントとし、各セグメントにループ各部の明暗に応じた明暗を与える、かつループ形状に応じてセグメントを屈曲させて、屈曲後のセグメントを合成してループ見本を形成するための、ループ見本作成手段と、編物をシミュレーションするように、ループ見本をループ形状決定手段のデータにより定めた位置に記憶して、複数のループを記憶するためのフレームメモリと、ループ形状決定手段のデータにより、下地のループとの重なりを表すマスクを作成するためのマスク作成手段と、前記マスクを用いて、前記フレームメモリに、下地ループの露出部を残すように、ループ見本を書き込むための描画手段と、前記フレームメモリの画像を表示するためのモニター、とを設けたことを特徴とする。

【0007】好ましくは、フレームメモリの画像をループのアスペクト比に応じて補正する手段を設けて、アスペクト比の補正後の画像をモニターに表示させる（請求項2）。また好ましくは、ループ見本を記憶するための手段を設けて、出現頻度の高いループ見本を記憶するようとする（請求項3）。さらに好ましくは、マネキンを表す画像を記憶するための第2のフレームメモリと、メッッシュマッピング手段とを設けて、第1のフレームメモリの編物のシミュレーション画像を、第2のフレームメモリの画像にメッッシュマッピングする（請求項4）。

【0008】ループ見本にはループ各部の明暗を反映させるので、例えば1つの糸に対して明暗の異なる複数の糸見本を糸見本記憶手段に記憶させ、ループ見本作成手段でループ各部の明暗に応じた糸見本を取り出すようにしても良い。あるいはまた糸見本記憶手段では、1つの糸に1つの糸見本を記憶させ、ループ見本作成手段で取り出した糸見本の明暗を変え、ループ各部の明暗に応じた値としても良い。ただし最初から明暗の異なる複数の糸見本を記憶させる方が、処理速度が速い。出現頻度の高いループ見本としては、例えば振りの無いループ等を基本ループとし、基本ループを記憶するようにする。

【0009】

【発明の作用】この発明では、編機用データを解析して、ループの形状を含め編物をシミュレーションする。まず編機用データから、各ループの糸の種類とループの形状、ループ各部の明暗を決定する。また1つの糸に対して明暗の異なる複数の糸見本を記憶し、ループを複数のセグメントに分割して、明暗に応じた糸見本を用いてセグメントを作成する。次にループ形状に応じてセグメントを屈曲させてループの変曲部を形成し、各セグメントを合成してループ見本とする。このようにして形成したループ見本は、糸の種類とループの形状及びループ各

4

部の明暗を反映したものである。編機用データの解析では、ループの重なり具合を分析し、下地のループとの重なりを表すマスクを作成する。このマスクを用いて、下地ループの露出部を新たに書き込むループ見本で消さないように、フレームメモリに書き込む。

【0010】ループのアスペクト比（縦横比）は一般に1ではなく、アスペクト比を無視してシミュレーションすると、実際の編物から遊離した形状となる。そこで好ましくは、フレームメモリのシミュレーション画像をループのアスペクト比に応じて補正し、アスペクト比を補正した画像をモニターに表示する。また好ましくは、シミュレーションに要する時間を短縮するため、出現頻度の高いループ見本を記憶し、次回からは作成済みのループ見本をスタンプして、ループ見本の作成回数をカットする。さらに好ましくは第2のフレームメモリにマネキンを表す画像を記憶させ、編物のシミュレーション画像を第2のフレームメモリのマネキン画像にメッッシュマッピングする。このようにすれば、シミュレーションでマネキンに編物を試着させることができる。メッッシュマッピング後の表示は、モニターを用いても良く、デジタルカラープリンタ等を用いても良い。

【0011】

【実施例】図1～図20に実施例を示し、図1にはニットデザインシステムのハードウェア構成を示す。図2～図10にループシミュレーションの前段のニットペイント処理を示し、図11～図17に実施例のループシミュレーション処理を、図18～図20にその後のメッッシュマッピング処理を示す。なお図2以下の表示は、図1のハードウェア構成をソフトウェアにより実効化したものである。

【0012】図1において、2はメインバスで、4はホストCPU、6は主メモリ、8はグラフィックCPUである。入出力としては例えばスキャナー10と、フロッピーディスクやハードディスクあるいは光磁気ディスク等の外部記憶12、デジタイザー14とスタイラス16、キーボード18等を用いる。グラフィックCPU8にはグラフィックバス20を接続し、バス20に一群のフレームメモリ22とワークメモリ23、アフィン変換処理部24、メッッシュマッピング処理部26を接続し、フレームメモリ22のデータを合成処理部28で合成し、グラフィックモニター30に表示する。

【0013】デザイン過程での編物のデータは、例えば1ループが1ピクセルの内部データとして、フレームメモリ22に記憶させる。同様に作成過程での水玉や直線、リピート模様等は、内部データ形式でワークメモリ23に記憶させる。アフィン変換処理部24では、ループのアスペクト比に応じて内部データを補正してモニターデータとし、補正結果をフレームメモリ22の他の領域に記憶させる。アフィン変換は、アドレス(x, y)のデータをアドレス(ax + by + c, dx + ey +

(4)

特開平7-70890

5

f) に変換することである。ここでcとfとはオフセット項を表し、フレームメモリ22とグラフィックモニター-30の表示アドレスとのオフセット補正に用いる。またアフィン変換で、座標xを $a x + b y$ に、座標yを $d x + e y$ に変換すると、水平垂直方向の伸縮のみでなく、斜め方向への変形や回転ができる。このことは、ループの向きがフレームメモリ22やグラフィックモニター-30の走査方向に対して斜めの場合も、アスペクト比の補正ができるることを意味する。しかしながら最低限必要な変換は、フレームメモリ22のアドレス(x, y)をアドレス(ax, by)に変換できることで、b/aがアスペクト比である。

【0014】 メッシュマッピング処理部26は、ループシミュレーション後のデータを仮想的にマネキンに試着させるためのものである。

【0015】 図2に、ニットペイント処理部を示す。図において、32はインターチャデータの格納領域、34は組織データの格納領域、36はジャガードデータの格納領域で、これらはフレームメモリ22に領域を指定して割り当てる。データは内部データとして記憶する。ここではインターチャと組織、ジャガードの3種類に分割したが、少なくとも2種類であれば良く、例えばインターチャかジャガードの模様と、組織の2種類とする。33は組織情報の追加部で、型紙データから目数の変化等の組織データを発生させ、インターチャデータ格納領域32に記憶した型紙の輪郭データに追加して、組織データの格納領域34に記憶させる。

【0016】 38, 40, 42は、アフィン変換手段24によりループサイズを補正した後のモニターデータの格納領域で、フレームメモリ22に領域を割り当てる。記憶する。38は補正後のインターチャデータの格納領域、40は補正後の組織データの格納領域、42は補正後のジャガードデータの格納領域である。これらのデータは合成処理部28で合成し、あるいは合成処理部28をバイパスして単独のデータのまま、グラフィックモニター30に表示する。合成処理部28では3画像の同時合成はできなくても良く、必要性の高い2画像の合成までがければ良い。モニター30での表示モードは、表示モード選択手段44で決定する。

【0017】 46は描画プロセッサーで、48はスタイルス16等の外部入力手段の入力座標を、モニター30上の座標に、座標アフィン変換するための座標アフィン変換手段である。実施例では、外部入力の位置は内部データに対応した内部座標として入力し、座標アフィン変換手段48でモニター座標(モニター30に対する座標)に変換する。

【0018】 50は編成方式入力で、キーボード18や外部記憶12等を用い、例えば編成の方法(インターチャ、ジャガード、組織)と、色糸の数、ゲージ及び目数等を指定する。52はループサイズ決定手段で、編成方

10

20

30

40

50

6

式とゲージや目数等に応じてループのサイズを決定する。54はサイズ入力で、採寸してキーボード18から数値し、あるいはスタイルス16から入力し、また外部記憶12に記憶済みの型紙データやスキャナー10から読み込んだ型紙データ等を入力する。56は型紙データ作成手段で、サイズ入力54からのデータに応じて型紙データを作成し、型紙外形決定手段58で型紙の外形を決定する。60は内部データ作成手段で、ループサイズ決定手段52からループのサイズを入力し、型紙外形決定手段58から型紙の外形を入力する。この結果、1ループが1ピクセルの内部データが発生し、内部データをデータ格納領域32, 34, 36等に記憶する。

【0019】 編物に対して模様や柄を入力するには、スタイルス16や柄描画データ入力62(外部記憶12に記憶させた既存の柄や模様、スキャナー10から読み取った既存の柄や模様)などを用い、入力位置をスタイルス16を用いてデジタイザー14で指定し、内部アドレスとして記憶すると共にモニターアドレスとして表示する。描画プロセッサー46は入力データを処理し、内部アドレスで指定されたピクセルに書き込む。一方グラフィックモニター30への表示では、デジタイザー14で指定した内部アドレスを、座標アフィン変換手段48でモニターアドレスに変換して表示する。

【0020】 なお、ループサイズ補正後のモニターデータを逆アフィン変換して内部データとはしないのは、逆アフィン変換による内部データの精度低下を防止するためで、編機データの基礎となる内部データを優先した。

【0021】 64は自動制御処理手段で、内部データを編機の編成データに変換するためのものである。組織データを1ピクセルにカラー情報として記憶させ、他のデータをオプション情報として記憶させた場合、内部データを編機データ(編成データ)に変換できることは、前記の特公平3-21661号公報により公知である。そこでインターチャ、組織、ジャガードの3種類のデータを特公平3-21661号公報のデータ形式に変換できれば、編機データに内部データを変換できる。そのため組織のデータはそのまま用い、インターチャ、ジャガードのデータからオプションラインのデータを生成させる。そしてこのデータを、前記の特公平3-21661号の自動制御処理手段で処理し、編機データに変換する。66は編機データのメモリ、68はフロッピーディスクで、編機データを記憶させる。もちろん編機データへの変換手法は任意である。

【0022】 図3に、型紙データと、内部データ、モニターデータの関係を示す。内部データ作成手段60で図の左側の型紙データが作成された場合、データ格納領域32, 34, 36には図の中央のように変形された内部データが記憶される。これは内部データでは1ループが1ピクセルに対応し、ループのアスペクト比が考慮されていないためである。次に内部データをアフィン変換

(5)

特開平7-70890

し、ループのアスペクト比を補正してモニターデータとすると、図の右側のようにモニター30に表示される。スタイルス16のカーソル位置をクロスマークで示すと、カーソル位置は内部アドレスで入力され、座標アフィン変換手段48により変換されて、グラフィックモニター30上にモニターアドレスで表示される。これらの結果、デザイナーは、内部データを考慮することなく、デザインできる。

【0023】図4に、ループサイズの補正とスタイルス16で指定したカーソル位置の座標アフィン変換のアルゴリズムを示す。最初に表示モード選択手段44で、インターチャ、組織、ジャガードのどのデータを修正するかを選択し、スタイルス16でカーソルの座標位置を内部アドレスで指定し、座標アフィン変換してモニターアドレスで、グラフィックモニター30に表示する。次いでスタイルス16等で描画領域を決定し、描画過程のデータは内部データとして例えばワークメモリ23に記憶し、アフィン変換してモニター30に表示する。デザイナーが描画を確認すると、描画プロセッサー46を用いて、データ格納領域32、34、36にワークメモリ23の内部データを入力する。描画過程のワークメモリ23の内部データは、アフィン変換手段24でモニターデータに変換し、データ格納領域38、40、42に転送し、グラフィックモニター30に表示する。

【0024】図5に、インターチャ、組織、ジャガードの3つのデータの合成を示す。図4のアルゴリズムで、内部データや内部アドレスは仮想化され、モニターデータやモニターアドレスのみが見えるので、図5以下では、ループサイズ補正後のモニターデータを示した。これらの図において、内部データや内部アドレスによるデザイン処理と、モニターデータやモニターアドレスによる表示系との2種類の処理があり、モニター上でデザイナーが処理を確認する都度、内部データが処理されるものとする。

【0025】内部データ作成手段60のオリジナル画像は、例えばインターチャ、組織、ジャガードの3種類、あるいはインターチャと組織等の2種類のデータとして格納領域32、34、36に書き込み、独立に描画プロセッサー46により修正する。このため、インターチャの模様をデザインするには、組織やジャガードの模様に考慮を払う必要はなく、インターチャ模様のみをデザインすれば良い。また組織のデザインでは、インターチャ模様やジャガード模様に考慮を払う必要はない。この結果、模様と組織とを別個にデザインでき、デザイン上の負担が小さくなる。インターチャやジャガード、組織の各データは、独立にコピー、移動、削除、修正ができる。例えばポケットの組織や組織柄のみを既存のデータからコピーでき、インターチャやジャガードの模様のみを既存のデータからコピーし、移動し、修正し、縮小拡大できる。

【0026】図5に示したように、ジャガードの模様が組織柄に対しどの位置にあるかは、組織データ格納領域34とジャガードデータ格納領域36の画像を合成すれば確認できる。同様にインターチャと組織柄との関係も、データ格納領域32、34の合成で確認できる。さらに各画像は、ループのアスペクト比を補正したモニターデータとして表示されるので、模様や柄の移動・コピー・縮小・拡大が容易になる。例えば図5の合成画面で、ジャガードのワンポイントマークを移動させる場合、ループのアスペクト比を補正していない内部データでは、実際の編物でのマークの位置をイメージするのは難しい。このためマークの移動やコピーは難しく、コピー後や移動後のマークが与えるイメージを理解するのも難しい。これに対してアスペクト比を補正したモニターデータでは、モニター30の表示形状は実際の編物の形状に相似し、実際の編物でのマークの位置が表示され、マークのイメージを直ちに理解できる。なお修正が終れば、3種類のデータを参照表等を用いて1種類のデータに変換し、自動制御処理手段64で編機の編成データに変換する。

【0027】図6に、描画プロセッサー46による、水玉やリピート模様、直線のサポートを示す。図において70はプロセッサー本体、72はズーム処理部、74は移動処理部、76はコピー処理部、78は水玉処理部、80はリピート処理部、82は直線処理部である。これらの各処理部72～82はプロセッサー本体70の助けを借りて各自のジョブを行い、指定したデータ格納領域32、34、36に対して書き込みを行う。ズーム処理部72のジョブは、入力した模様や柄を縮小あるいは拡大することで、移動処理部74のジョブは、入力した模様や柄を移動することである。またコピー処理部76のジョブは、入力済みの模様や柄をデータ格納領域32、34、36の他の部分にコピーすることである。これらの処理部72、74、76は通常のペイントシステムで周知で、容易に実現できる。図10に、水玉処理、リピート処理、直線処理のアルゴリズムを示す。

【0028】図7に、水玉処理部78での処理を示す。水玉模様をデザインする場合、水玉をスタイルス16で描画し、1つずつコピー処理部76でコピーするのは面倒である。そこで図7の右側のように、水玉の要素となる円A、B、C、Dを入力し、これらの水玉A、B、C、Dを一括して移動あるいはコピーするための基準点Eを入力する。基準点Eと個々の水玉A～Dとの間の直線は、水玉A～Dが基準点Eと関係していることを表すためのラインで、制御棒と呼ぶ。水玉A～Dや基準点E等のデータは内部データとしてワークメモリ23に記憶し、アフィン変換して表示し、カーソル位置も内部アドレスで指定し、座標アフィン変換して表示する。モニター30上で水玉模様をコピーする位置を基準点Eの位置として指定すると、基準点Eの内部アドレスを基に、水

(6)

9

玉模様をコピーする。実際にコピーするのは、水玉A, B, C, Dのみで、基準点Eや制御ラインはコピーしない。

【0029】図8に、リピート処理部80での処理を示す。図のC, D, E, Fはリピート領域を指定するための点で、円A, Bはリピート領域と編物のアウトラインとの交点である。点A, A2, A3, A4は1単位の模様が占める領域を指定するための点で、この場合点Aと点A2間の幅が、模様の単位の幅となる。そして点A, A2, A3, A4で指定された領域に、1単位の模様をスタイルス16等で入力し、リピート領域(C, D, E, F)に囲まれた部分の模様をピックアップし、リピート処理部80で単位模様をリピート領域内に繰り返しコピーする。この結果、模様を1単位分入力することのみで、正確に繰り返しコピーできる。Gは移動ポイントで、リピート領域内の1点をスタイルス16等で指定し、クリックする。次にスタイルス16を適宜の位置に移動すると、模様をその位置に移動し、あるいは模様をその位置にコピーする。なおリピート処理部80でも、作成過程のデータやカーソル位置は内部データと内部アドレスで処理し、データ格納領域32, 34, 36への書き込みも内部データ形式で行う。一方モニター30への表示は、アフィン変換を施し、モニターデータとモニターアドレスで処理する。

【0030】図9に直線処理部82の動作を示す。スタイルス16でまっすぐな線を引くことはそれ自体として難しく、アリエーシングが最も少ない規則的な線を指定することはさらに難しい。そこでワークメモリ23等に対して角度の調整ポイントを2点で指定し、この間を直線で補間する。角度調整ポイント間のデータはベクトルデータで、このベクトルデータに最も近い直線、即ち最もアリエーシングの小さな直線を発生させる。直線の移動は、リピート処理部80での移動ポイントの処理と同様で、直線上の1点をクリックし、これを指定の点まで移動すると、新たな直線を発生し、あるいは直線をその点まで平行移動できる。なお直線処理部82でも、ワークメモリ23やデータ格納領域32, 34, 36での処理は内部データで行い、グラフィックモニター30にはアフィン変換を施したモニターデータを表示する。

【0031】図11～図17に、実施例のループシミュレーション処理部を示す。図11にアーキテクチャーを示し、図1のハードウェア構成にソフトウェアを加えて実現する。図11において、90はループ形状決定手段で、自動制御手段64の編機用データをループ毎に解析し、各ループに用いる色糸の種類とループ形状、ループ各部での明暗、下地のループとの重なり具合を解析する。ループ形状の解析はコースに沿って1ループずつを行い、1コースの解析が終了すると次のコースを解析する。

【0032】92は糸見本記憶手段で、例えば16種の

特開平7-70890

10

色糸を用いる場合、1種類の色糸に対して10個程度の糸見本を記憶する。各糸見本の相違は明暗であり、糸見本自体の明暗と周囲の明暗とを変え、糸自体の明暗(表目、裏目等の違いとループ内の位置による明暗)を表現し、糸のエッジを強調あるいはエッジをぼかす。処理速度が遅くても良い場合、1つの糸に1つの糸見本を記憶させ、切り出したセグメントの明暗をループ各部の明暗に応じて変えて良い。糸見本には図14に示すように、周囲に凹凸を与えてけば表現し、糸のよりを表現する線を書き込む。94は糸見本切り出し用のマスクの作成部で、例えば図14に示すような円形のマスクを作成し、マスクの周囲の遷移領域でマスクの値を徐々に変化させ、ソフトなマスクとする。マスクの半径は糸見本から切り出すセグメントの長さに応じて変化させ、セグメント長が一定の場合糸見本をそのまま用いれば良く、マスクは不要である。96はスプライン変換部で、ループの基端と先端との屈曲部で、セグメントを屈曲させる。全てのセグメントにスプライン変換を施しても良いが、ループの基端と先端との間はほぼ直線に近く、この部分でスプライン変換を省略する。98は合成処理部で、ループの各セグメントを合成し、1つのループ見本とする。

【0033】100は下地ループ用のマスク作成手段で、1コース前のループに対する新たなループの重なり具合を表現するため、マスクを作成する。102はループ見本のメモリで、例えばワークメモリ23等に領域を指定して実現し、作成したループ見本を一時的に保存すると共に、出現頻度の高い基本ループの見本をそのループの処理が終了した後も保存する。46は前記の描画プロセッサーで、104はループシミュレーション画像のフレームメモリである。フレームメモリ104は、図1のフレームメモリ22に領域を割り当てて実現する。描画プロセッサー46では、1コース下のループのデータをフレームメモリ104から呼び出し、下地のループ(1コース前のループ)が露出する部分を下地ループのマスクでマスクして新たに書き込むループ見本から保護し、メモリ102からループ見本を呼び出して、フレームメモリ104に書き込む。

【0034】ループの形状は一般に正方形グリッドにフットせず、アスペクト比は1ではない。そこで実際の編物をシミュレーションするため、アフィン変換手段24を用いてアスペクト比を補正する。最低限必要な変換は、フレームメモリ104でのアドレス(x, y)の画像を、アドレス(ax, by)の画像に変換することで、必ずしもアフィン変換である必要はない。106はアフィン変換手段24によりループサイズを補正した後の画像のフレームメモリで、フレームメモリ22の一部に領域を指定して実現する。そしてループシミュレーション後にアフィン変換を施したフレームメモリ106のデータを、グラフィックモニター30に表示する。

(7)

特開平7-70890

11

【0035】図12～図17により、ループシミュレーションでの処理を示す。処理のメインアルゴリズムを図12に示す。自動制御処理手段64からの編機用データを解析すると、各ループの基礎中心位置とループの先端中心位置、並びに用いた糸の種類や、表目か裏目その他の編成データ、またループ各部の明暗等が判明する。そこでこれに応じて、ループ見本を作成する。

【0036】図13～図16に、ループ見本の作成過程を示す。ループ形状決定手段90では、ループ見本を作成済みかどうかを検査し、作成済みであればメモリ102からループ見本を呼び出して用いる。またループ見本が未作成でしかも振りの無い基本ループの場合、作成したループ見本をメモリ102に記憶させる。新たにループ見本を作成する場合、ループをその曲率や明暗に応じて複数のセグメントに分割し、セグメントを糸見本記憶手段92に記憶した糸見本から切り出す。この場合ループ見本は全て新たに作り出す必要はなく、記憶済みのループ見本と共に部分は兼用し、処理速度を向上させるのが好ましい。例えば基本ループの見本はループ毎に1回しか作成しないので、ループ見本の作成の大部分は振りのあるループである。振りのあるループでも、ループの先端部と基礎部との形状は基本ループと同じで、違うのはその間である。そこで先端部と基礎部は基本ループの見本からコピーし、その間のセグメントのみを新たに作成し、振りのあるループの見本とすれば良い。

【0037】図14に示すように、糸見本記憶手段92には、糸自体の明暗と周囲とのコントラストを変えて、1つの糸に対して10種類程度の糸見本を記憶する。また各糸見本には、糸の周辺のけばや糸のより、糸の太さ、色等を記憶させてる。セグメントの長さをループ形状決定手段90で決定するとマスクの直径が定まり、マスク作成部94からセグメントの切り出し用マスクを取り出す。マスクは図14の右側に示したように例えば円形で、その周辺には遷移領域があり、周辺をソフトにしている。マスクの値Zが1で糸見本をそのまま切り出し、0で糸見本を切り出さないものとすると、マスクの中央部は透明でZが1で、マスクの外部ではZが0となり、周辺の遷移領域でZは1から0へと徐々に減少する。このようなマスクを用いて糸見本を切り出すと、セグメントは図14の下に示したように、中央部は糸見本のデータをそのまま切り出し、周辺の遷移領域（図の破線）では糸見本のデータをソフトに切り出すことになる。

【0038】ループの基礎部と先端部には曲率があり、その間は直線に近いので、基礎部と先端部とをスプライン近似し滑らかに屈曲させる。そして各セグメントを合成処理部98で合成し、1つのループ見本とする。この過程を図15に示すと、表目／裏目等の種類によってループの各部には明暗があり、先端部は曲率中心C1を中心に半円状をなし、基礎部は双曲線の焦点C2を中心に

10

12

屈曲している。そこで切り出した各セグメントを曲率に応じてスプライン変形させ、屈曲させる。実施例ではループ見本の全てのセグメントにスプライン変換を施したが、基礎と先端との間は直線近似でも良い。

【0039】各セグメントを単純に結合すると、セグメントとセグメントとの接続部でエッジが生じる。そこで図14に示したマスクにより、セグメントの両端部をソフトにし、これを重ね合わせて合成する。この過程を図16に示す。セグメントS2とセグメントS3との境界部で各セグメントの明度は滑らかに低下し、これらを重ね合わせると滑らかにセグメントを結合できる。

【0040】ループ見本を発生させると、図12のメインプログラムに戻り、マスク作成手段100を用いて、下地ループへのマスクを作成する。このマスクを図17に示す。下地ループのマスクは下地のループが新たなるループによって覆われず露出する部分をマスクして保護したもので、マスクの値Zは、図17の右側に示したように、マスクした部分で1、マスクしない部分で0で、その間を滑らかに接続する。マスクを形成すると、フレームメモリ104から下地ループの画像データを取り出し、メモリ102からループ見本を取り出し、マスクを用いて描画プロセッサー46で両者の重なりを処理し、フレームメモリ104に書き戻す。

【0041】フレームメモリ104の画像データは編物をそのループを含めて詳細にシミュレーションしたものであるが、アスペクト比が補正されていない。そこで1ループの解析が終了する都度、アフィン変換手段24によりループのアスペクト比を補正し、フレームメモリ106に記憶しモニター30に表示する。このためループシミュレーションでは、解析の終わる毎に1ループずつ表示ループの数が増え、徐々に編物の姿が現れてくる。このようにすれば、編物を実際に編まずに、またループの見本をスキャナー等で取り込みますに、多種多様のループに対して、明暗や周囲とのコントラスト、ループの形状、重なり具合等を含めて、正確にシミュレーションすることができる。さらに編機用データが編めないデータ、例えば目が落ちるデータの場合、編機用データを基にループをシミュレーションするので、モニター30から編めないデータであることを検出できる。その場合モニター30には、1ループずつ順にループが姿を表すので、シミュレーションの間モニター30を監視していれば、編めないループを見落とすことがない。

【0042】図18～図20に、メッシュマッピングによる、マネキンへのループシミュレーション画像の試着を示す。図18において、110はメッシュマッピング用のプロセッサーで、106はループサイズ補正画像のフレームメモリ、112はマネキン形状のフレームメモリ、114はマネキンの明暗画像のフレームメモリ、116はマネキンへのマスクを表すフレームメモリである。また46は前記の描画プロセッサーで、118はメ

50

(8)

特開平7-70890

13

ツッシュマッピング処理の結果を記憶するためのフレームメモリである。

【0043】 メッシュマッピングでは、ループシミュレーション後のアスペクト比を補正した画像（フレームメモリ106の画像）を用い、マネキンの形状やその明暗、マスク等を用い、各々フレームメモリ112, 114, 116に記憶する。次にスタイルス16を用いて、マネキン画像とループシミュレーション画像の主要ポイントをマッピングする（図18での0マーク）。マッピングしたポイントの間をフレームワーク（骨格）で接続すると、例えば図19に示すように、ループシミュレーション画像での正方形の領域S1, S2, S3, S4は、マネキン画像では図の右側のように変形する。そこでフレームワークで囲まれた部分毎に、ループシミュレーション画像とマネキン画像を呼び出し、編物をマネキンに試着させることで生じるピクセル数の増加や減少を補うように補間し、図の実線から破線のように順に走査し、描画プロセッサー46で変形した画像を発生させる。この画像は、ループシミュレーション画像の値をP1, マネキン画像の値をP2, マネキンの明暗画像の値をP3, マネキンのマスク画像の値をZとすると、

$P1 \cdot P3 \cdot Z + (1-Z) \cdot P2$
となり、この値をフレームメモリ118に書き込み、モニター30に表示する。そして例えばフレームメモリ118の画像をカラープリンタで出力すればプレゼンテーション用のサンプルが得られる。

【0044】 このようにすれば、実際に編物を編むことなしに、デザインした編物をマネキンに試着させることができます。またマネキンの形状と明暗とを反映し、平面状の編物をマネキンに試着させることによる立体感を表現し、それに応じて編物にひだ等の伸縮を発生させることができる。

【0045】

【発明の効果】 この発明では、以下の効果が得られる。

- 1) ループ見本を実際に編んで作成せずに、各ループを正確に表現するようにシミュレーションできる（請求項1～4）。
- 2) 各ループを、その形状、屈曲の具合い、ループ各部の明暗、他のループとの重なり具合いを含めてシミュレーションできる（請求項1～4）。
- 3) 編機用データをチェックし、実際に編めるデータかどうかを確認できるようにする（請求項1～4）。
- 4) ループのアスペクト比を補正し、実際の編物に近い形状でシミュレーションできる（請求項2, 4）。
- 5) シミュレーション時間を短縮する（請求項3）。
- 6) シミュレーションした画像をマネキンに試着させ、立体感のあるシミュレーションを行う（請求項4）。

【図面の簡単な説明】

【図1】 実施例のニットペイントシステムのブロック図

14

【図2】 実施例のニットペイントシステムの要部詳細ブロック図

【図3】 ニットペイントでの、型紙データと内部データとモニターデータとの関係を示す図

【図4】 ニットペイントでの、座標変換を示すフローチャート

【図5】 ニットペイントでの、インターフェイ、ジャガード、組織柄の3枚のデータの関係を示す図

【図6】 図2の描画プロセッサーの内部ブロックを示す図

【図7】 実施例での水玉処理を示す図

【図8】 実施例でのリピート処理を示す図

【図9】 実施例での直線処理を示す図

【図10】 水玉、リピート、直線の3つの処理の制御フローチャート

【図11】 ループシミュレーション処理部のブロック図

【図12】 ループシミュレーションのアルゴリズムを示すフローチャート

【図13】 ワークの発生アルゴリズムを示すフローチャート

【図14】 ループシミュレーションでの糸見本の切り出しを示す図

【図15】 セグメントからワークへの合成を示す図

【図16】 ワークの合成での、セグメントの重ね合わせを示す図

【図17】 ループシミュレーションでの下地ループへのマスクを示す図

【図18】 メッシュマッピング処理部のブロック図

【図19】 メッシュマッピングでの画像の変形を示す図

【図20】 マネキン処理のアルゴリズムを示すフローチャート

【符号の説明】

2 メインバス

4 ホストCPU

6 主メモリ

8 グラフィックCPU

10 スキャナー

40 12 外部記憶

14 デジタイザー

16 スタイラス

18 キーボード

20 グラフィックバス

22 フレームメモリ

23 ワークメモリ

24 アフィン変換処理部

26 メッシュマッピング処理部

28 合成処理部

50 30 グラフィックモニター

(9)

特開平7-70890

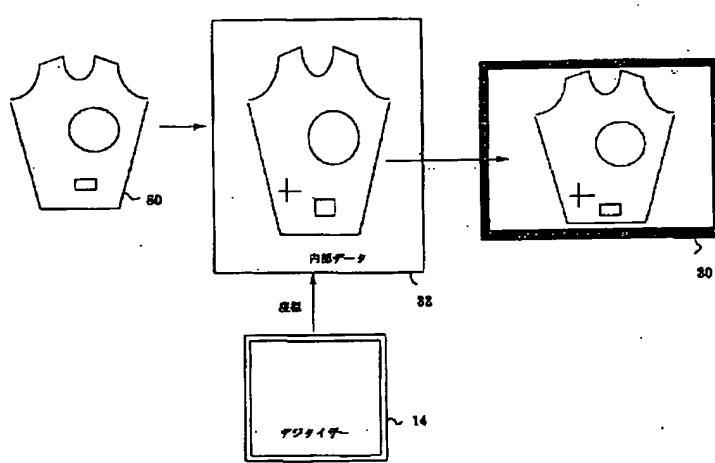
15

3 2 インターシャデータ格納領域
 3 3 組織情報追加部
 3 4 組織データ格納領域
 3 6 ジャガードデータ格納領域
 3 8 補正後のインターフェータ格納領域
 4 0 補正後の組織データ格納領域
 4 2 補正後のジャガードデータ格納領域
 4 4 表示モード選択手段
 4 6 描画プロセッサー
 4 8 座標アフィン変換手段
 5 0 編成方式入力
 5 2 ループサイズ決定手段
 5 4 サイズ入力
 5 6 型紙データ作成手段
 5 8 型紙外形決定手段
 6 0 内部データ作成手段
 6 2 柄描画データ入力
 6 4 自動制御処理手段
 6 6 編機データのメモリ
 6 8 フロッピーディスク
 7 0 プロセッサー本体

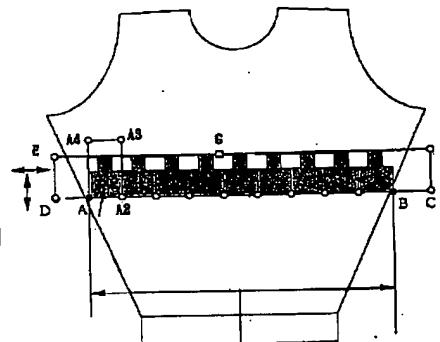
16

7 2 ズーム処理部
 7 4 移動処理部
 7 6 コピー処理部
 7 8 水玉処理部
 8 0 リピート処理部
 8 2 直線処理部
 9 0 ループ形状決定手段
 9 2 糸見本記憶手段
 9 4 糸見本切り出しマスク作成部
 10 9 6 スプライン変換部
 9 8 合成処理部
 10 0 下地ループマスク作成手段
 10 2 ループ見本メモリ
 10 4 ループシミュレーション画像のフレームメモリ
 10 6 ループサイズ補正画像のフレームメモリ
 11 0 マッピングプロセッサー
 11 2 マネキン形状のフレームメモリ
 11 4 マネキンの明暗画像のフレームメモリ
 11 6 マネキンマスクのフレームメモリ
 20 11 8 マッピング結果のフレームメモリ

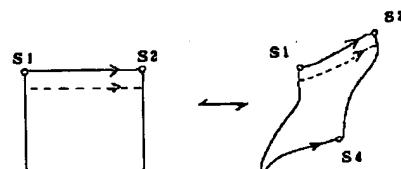
【図3】



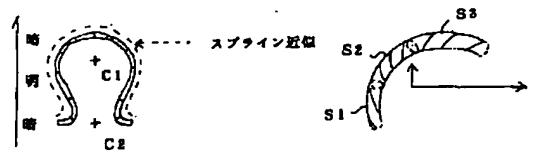
【図8】



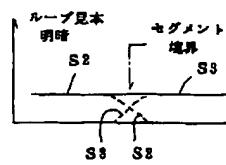
【図19】



【図15】



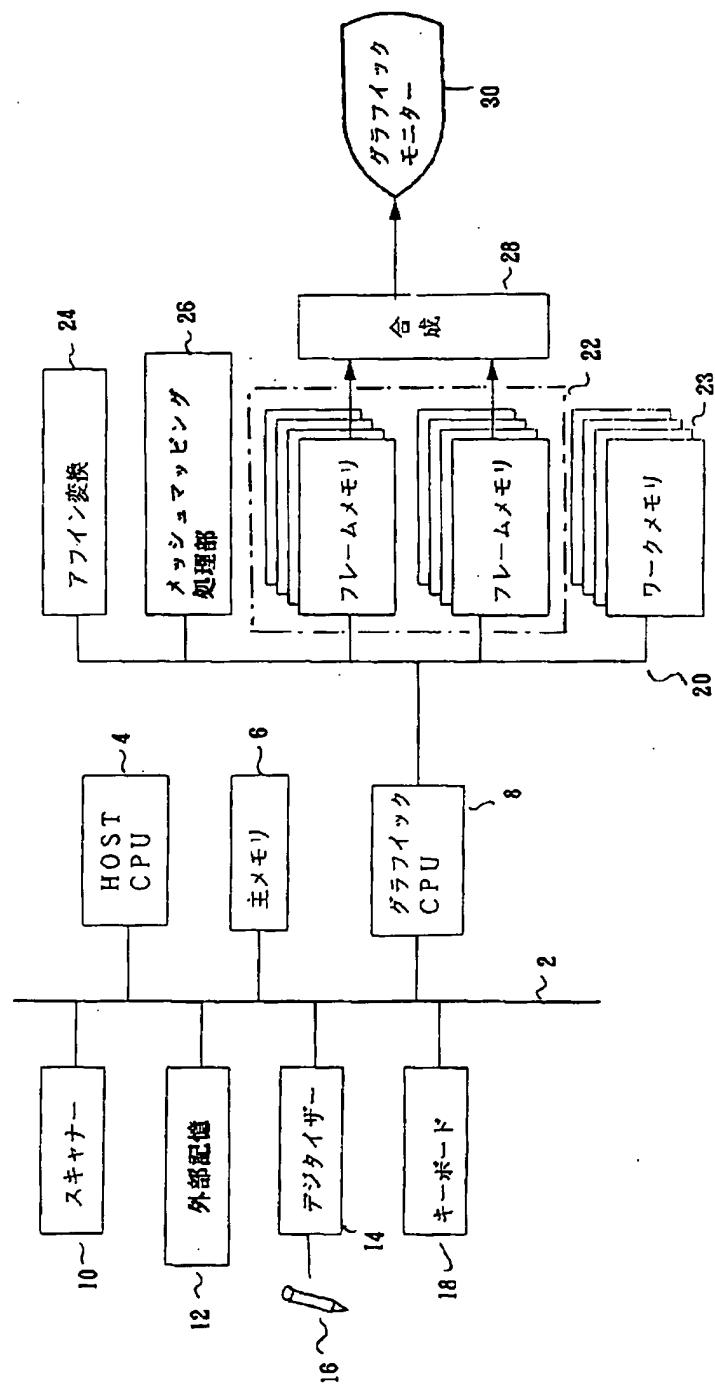
【図16】



(10)

特開平7-70890

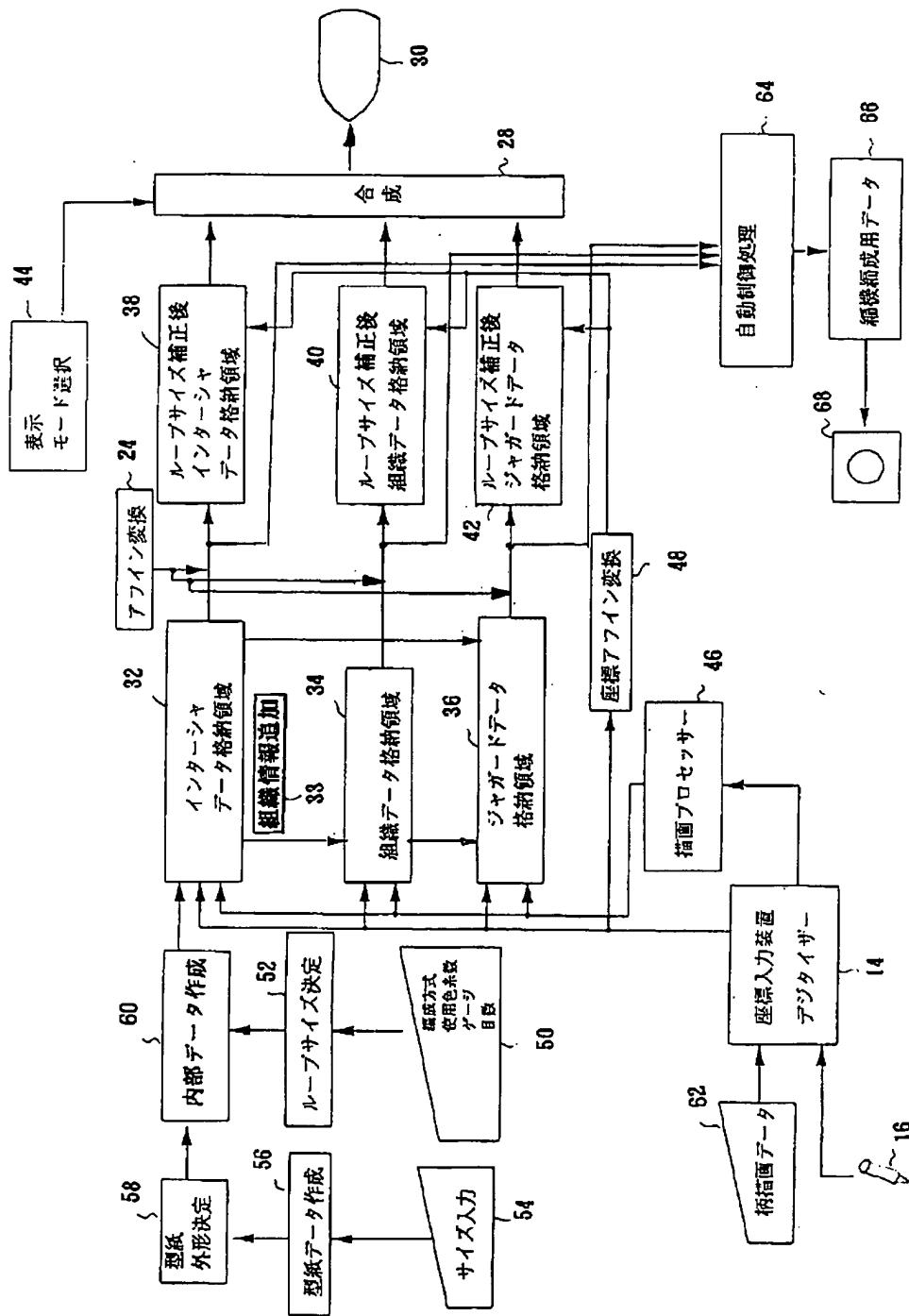
【図1】



(11)

特開平7-70890

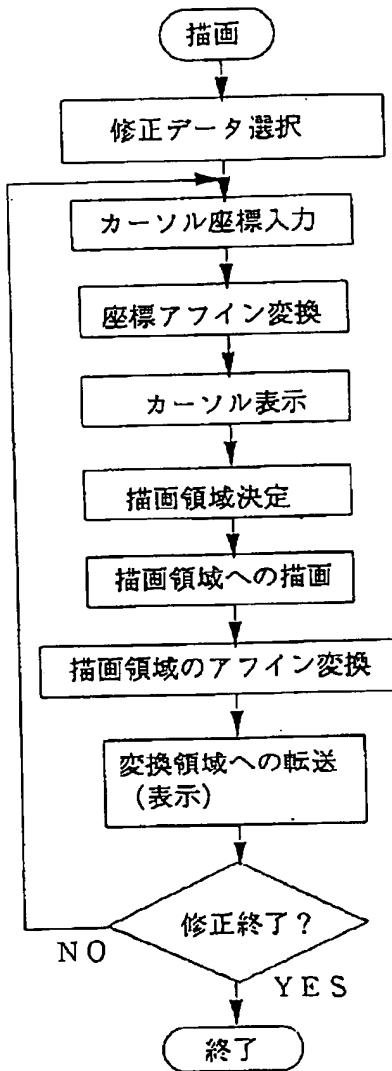
【图2】



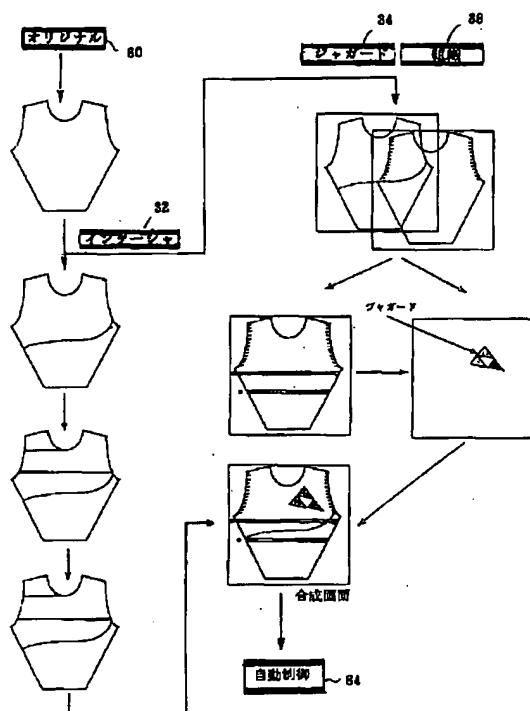
(12)

特開平7-70890

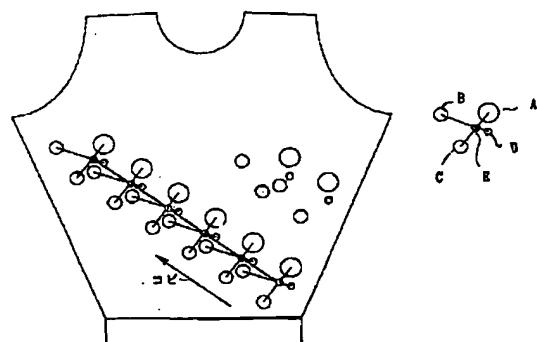
【図4】



【図5】



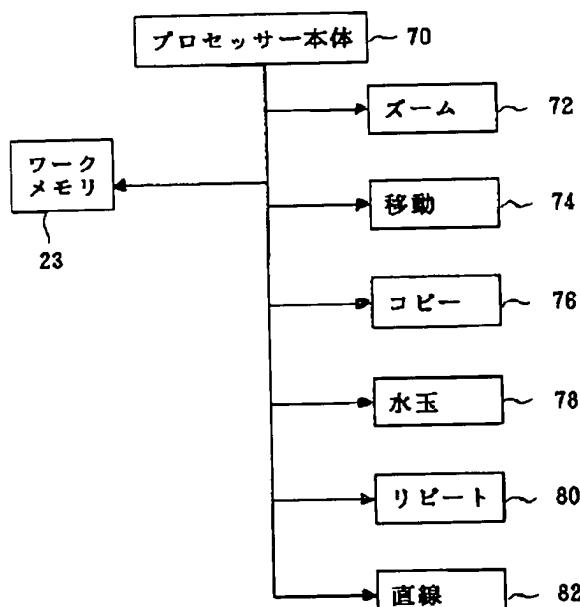
【図7】



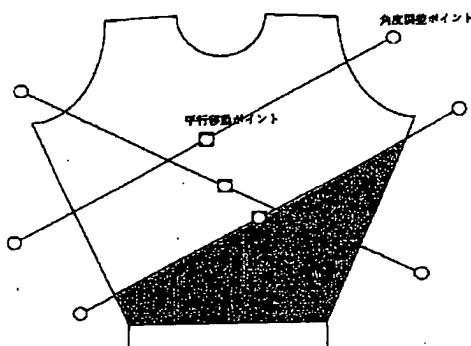
(13)

特開平7-70890

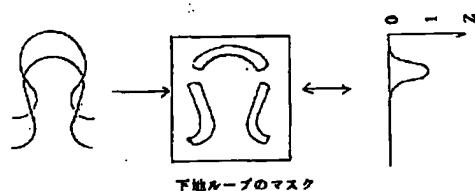
【図6】



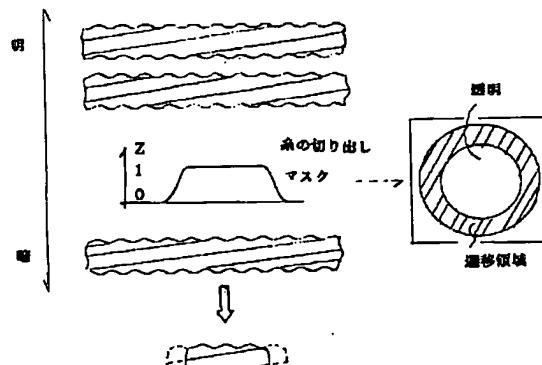
【図9】



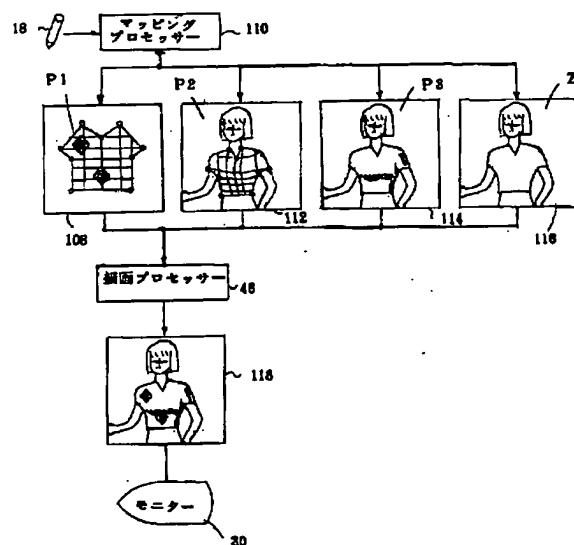
【図17】

46

【図14】



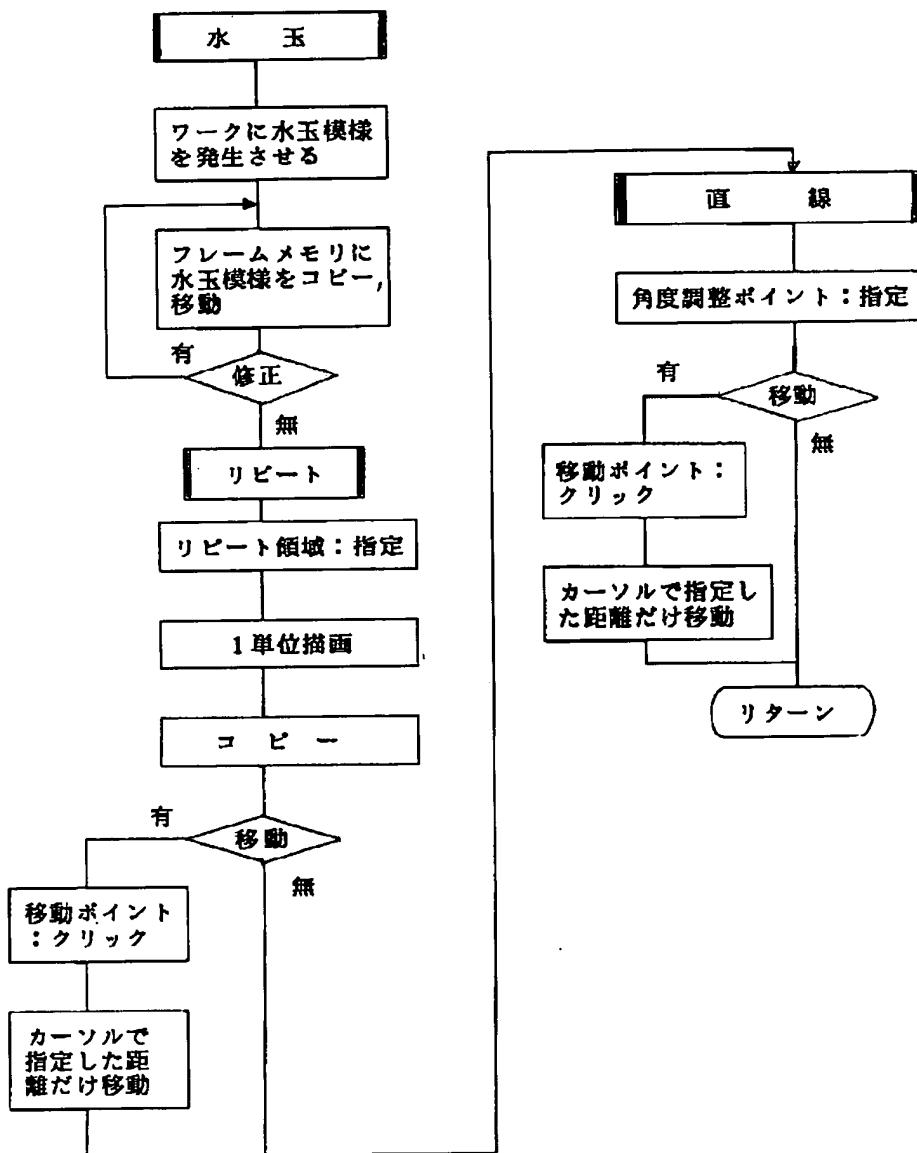
【図18】



(14)

特開平7-70890

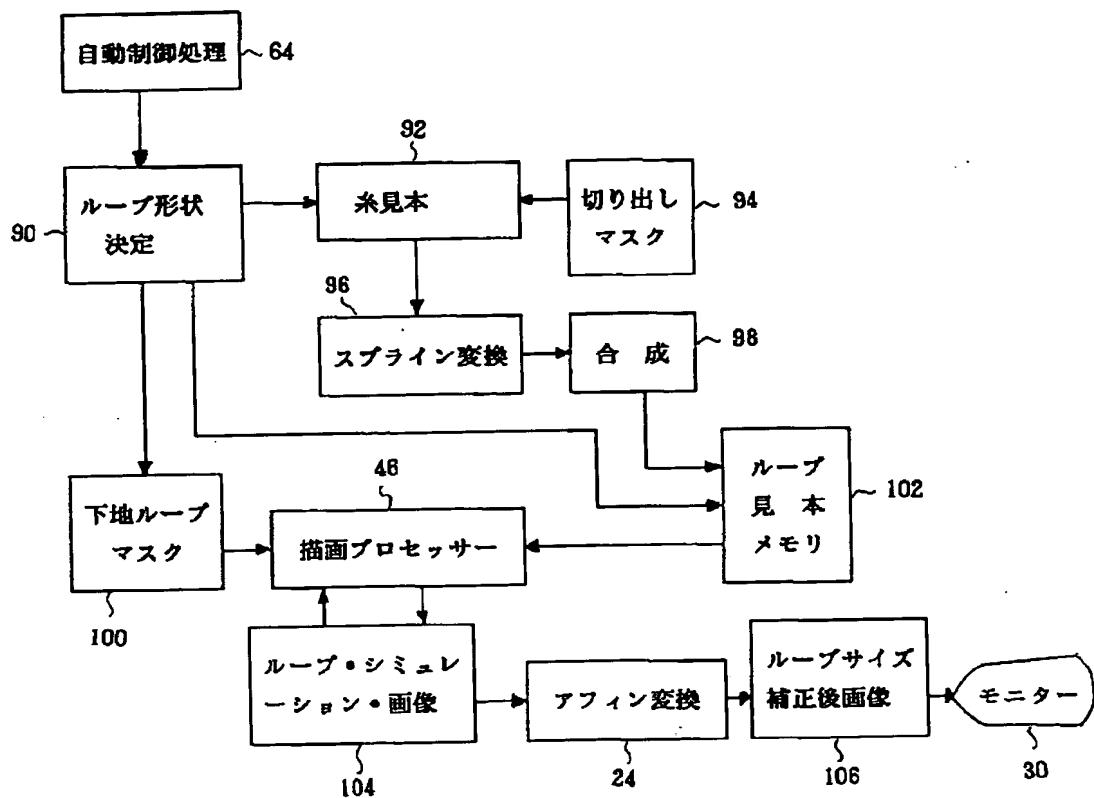
【図10】



(15)

特開平7-70890

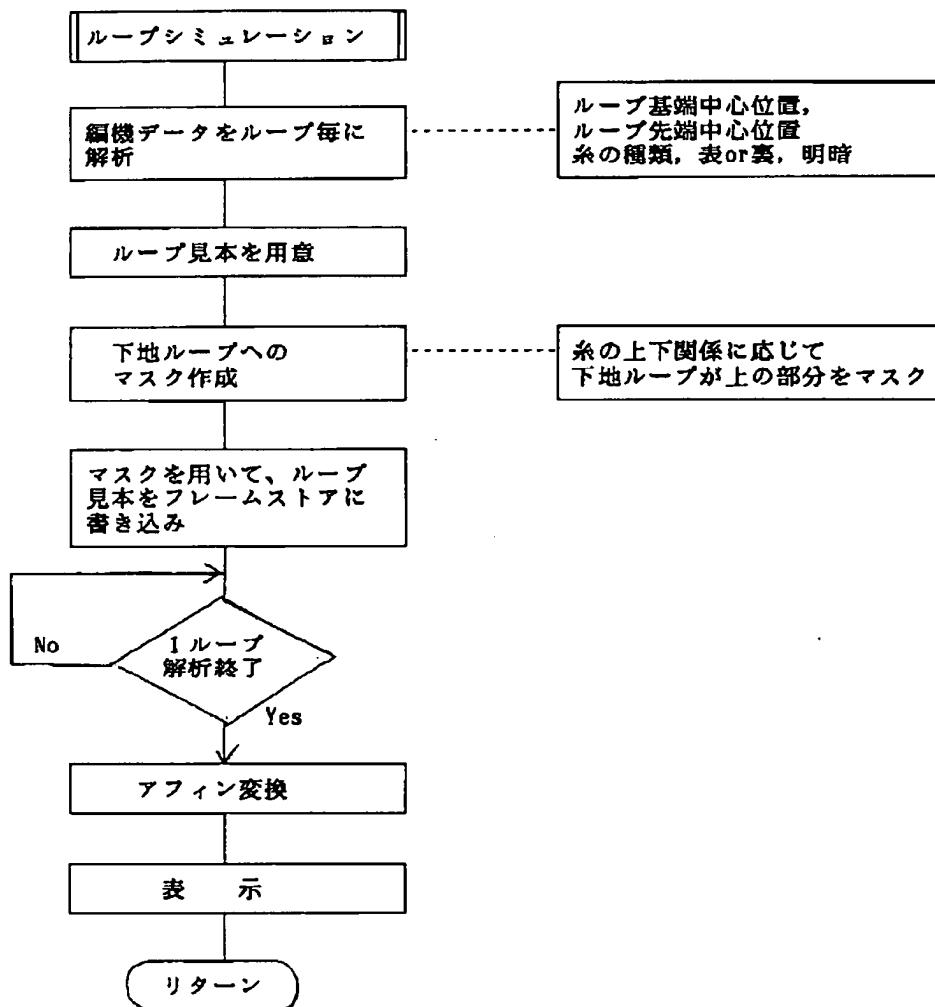
[図11]



(16)

特開平7-70890

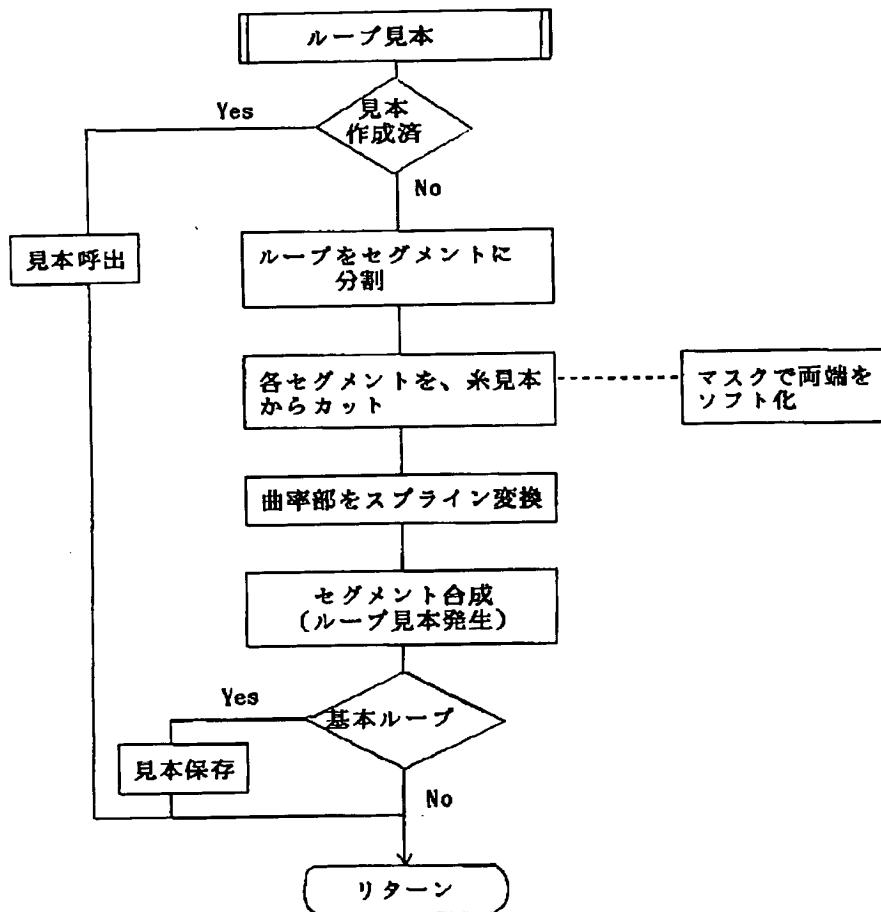
[図12]



(17)

特開平7-70890

【図13】



(18)

特開平7-70890

【図20】

